Realizing throughput guarantees in a differentiated services network *

Ikjun Yeom
A. L. Narasimha Reddy
Dept. of Elec. Engg.
Texas A & M University
College Station, TX 77843-3128

Abstract

This paper discusses techniques for achieving desired throughput guarantees in the Internet that supports differentiated services framework. Diff-serv framework proposes using different drop precedences to achieve soft service guarantees over Internet. However, it has been observed that the drop precedences by themselves cannot achieve the desired target rates because of the strong interaction of the transport protocol with packet drops in the network. This paper proposes and evaluates a number of techniques to better achieve the throughput guarantees in such networks. The proposed techniques consider (a) modifying the transport protocol at the sender, (b) modifying the marking strategies at the marker and (c) modifying the dropping policies at the router. It is shown that these techniques improve the likelihood of achieving the desired throughput guarantees and also improve the service differentiation.
Keywords : Differentiated service, Throughput guarantee, Quality of service, RED, TCP

1 Introduction

Service guarantees over networks have received wide attention recently with the increased use and need for mechanisms for delivering audio and video across networks. Different multimedia applications require different guarantees for delivery. Video-on-demand applications can tolerate large delays and primarily require throughput guarantees. Video-conferencing and IP telephony require delay guarantees as well as throughput guarantees. Much work has been done in designing multimedia applications that can tolerate delay and throughput variations by adjusting the quality of the transmitted stream. Simultaneously research is being carried out in providing service guarantees in networks. This paper deals with the issue of providing throughput guarantees in networks and how the application transport protocol needs to be adopted to take advantage of these guarantees.

Many scheduling approaches have been studied for providing service guarantees. These include variants of fair scheduling [12, 13, 14, 15] and priority scheduling [16, 1]. Recently, there has been a push to minimize the amount of work that needs to be done in a router to provide guarantees. Diff-serv framework is a proposal to provide service guarantees over networks by providing different drop preferences [2, 3, 4]. In this framework, the routers at the edge of the network monitor and mark packets of flows (individual or aggregated). The packets of a flow that obey the service profile are marked IN (in profile) and the packets that are beyond the service profile are marked OUT (out-of-profile). The network gives preference to IN packets while dropping OUT packets disproportionately at the time of congestion. The router doesn't distinguish between packets of individual flows and can use FIFO style scheduling mechanisms. This preferential drop mechanism is expected to provide better throughput for IN packets than OUT packets. With appropriate network provisioning, it is expected that this could result in bandwidth guarantees as long as the IN traffic doesn't exceed the link capacities in the network. Without other reservation mechanisms or route pinning mechanisms, the edge routers can't anticipate how all the flows get routed through the network. Hence, it is possible that occasionally the IN traffic may exceed the capacity on a particular link resulting in loss of guarantees. Figure 1 shows the different elements of a network. The realized throughput is a result of the interaction of the actions of the routers/switches inside the network, the sender, the marker and the interaction among the different flows.

The realized throughput is a result of the combina-

*This work was supported in part by a Texas ATP grant and by an NSF Career Award

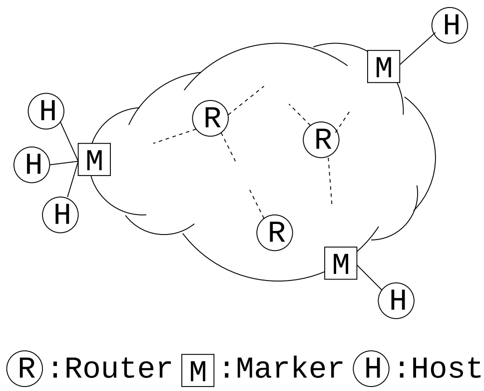


Figure 1: Network elements

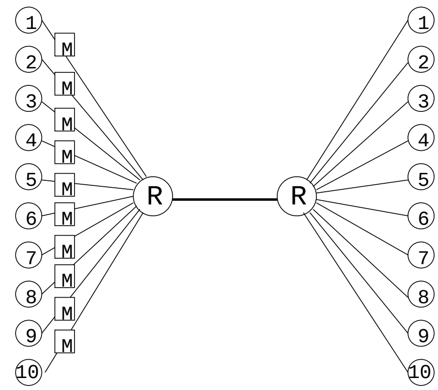


Figure 2: Network topology for simulations.

tion of dropping policy of the network and the policy of the transport protocol in how it reacts to these drops. TCP reacts to congestion by halving the congestion window and increases the window additively when packets are delivered successfully. Exponential decrease (halving the congestion window) is required to avoid congestion collapse [8] and TCP treats a packet drop as an indication of congestion. When unmodified TCP reacts to an OUT packet drop by halving its congestion window, it may not protect its reservation rate.

This paper studies this interaction between the transport protocol and the differentiated drop policies of the network in realizing the reserved throughputs. We propose a number of mechanisms to better realize the target rates. The proposed mechanisms can be classified into modifications to the transport protocol and modifications to the marking and dropping policies. Specifically, the paper makes the following contributions: (1) proposes new schemes for improving the realization of reserved rates in a differentiated services network, (2) evaluates the proposed schemes through simulations to show that the new schemes provide better realization of throughput guarantees, (3) evaluates the sensitivity of the throughput guarantees to different parameters such as round-trip-times (RTTs), (4) studies the impact of aggregation on the results and (5) proposes new quantitative measures for evaluating the various schemes.

2 Background

Let r_i denote the reserved rate of a flow i and C represent the capacity of a bottleneck link in the network. If a number of flows pass through this link, then $\sum_{i=1}^{n} r_i$ of the link capacity is allocated for IN packets, where n is the number of flows going through the link. The excess bandwidth at this link is then given by

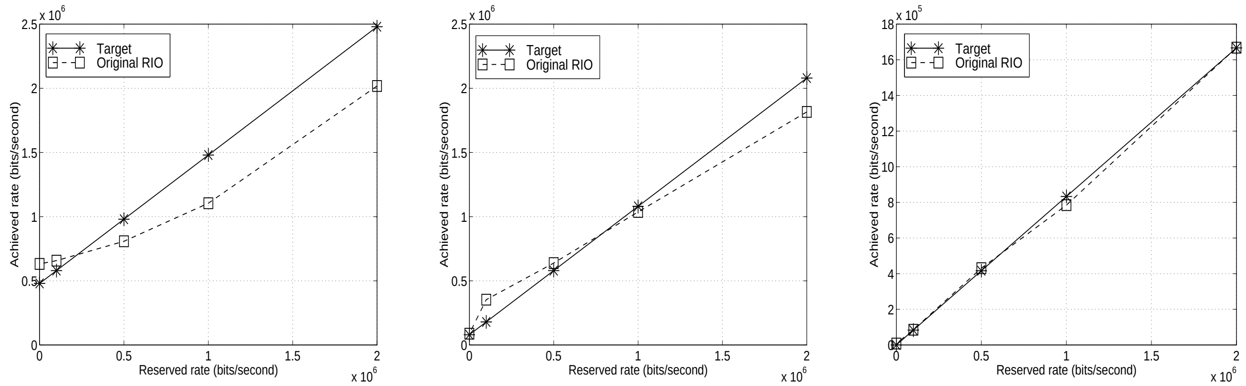
$$e = C - \sum_{i=1}^{n} r_i. \tag{1}$$

This excess bandwidth at the link beyond the allocated/reserved capacity can then be shared by all the flows. This excess bandwidth can be shared in many different ways. Sharing proportional to the reserved rates and equal sharing are two of the logical choices. We will deal with equal sharing of the excess bandwidth in this paper. Equal sharing allows flows without any reservation to continue receiving some service while proportional sharing may deny them of such service. Hence, a flow's target rate is given by

$$t_i = r_i + e/n. \tag{2}$$

The marking, dropping schemes along with the transport protocol's reaction to congestion determine how closely the flow can realize the target rate. To understand the dynamics of this interaction, we conducted several simulation experiments using the ns-2 [9] simulator.

Figure 2 shows a simple network topology that enables studying this interaction. Sources 1,2,...10 are TCP-Reno sources. The marker uses a sliding window leaky bucket marking strategy proposed in [4]. The router uses RED parameters 20/40/0.5 for the OUT packets and 50/100/0.02 for the IN packets. The reserved rates for each flow are shown in the figure. The total allocated bandwidth is 7.2 Mbps. The link bandwidth is set at 12 Mbs, 8Mbs and 6 Mbs in three different experiments to simulate allocations of 60%, 90%, and 120% of capacity. All the flows are assumed to have the same RTTs of 40 ms and run for 30 seconds. The results of the simulation are shown in Figure 3.



(a) 60%

(b) 90%

(c) 120%

Figure 3: Realized rates at different subscriptions

The realized throughput is considerably different from the target rates for most of the flows. The flows with smaller targets exceeded their target rates and the flows with larger targets did not achieve their target rates. A similar result was observed in [2]. TCP reacts to a packet loss by halving its congestion window. TCP then slowly increases the congestion window on successful transmission of packets. This additive increase of congestion window after a packet loss results in different recovery times to regain the congestion window for different flows. A flow with a larger rate takes longer time to reach its original rate compared to a flow with a smaller rate. Hence, the smaller flows realize higher rates.

Why did packet loss occur even when the allocated capacity is below the link capacity? First, the best effort flows may transmit data above the excess bandwidth available. Second, TCP flows continue increasing their transmission rates even after reaching their reserved rates until a packet drop. Third, bursty arrival may cause some packets to be dropped even when the combined sending rate is lower than the link capacity. For each flow, the loss in bandwidth from the target rate is a function of the number of packets of that flow dropped and the recovery time (to the original sending rate) as a result of a packet drop. In the following sections, we will look at methods that affect (a) the dropping rate and (b) the recovery time as a result of a packet drop. The dropping rate can be affected by the marker, the network, the sender (burstiness of the sender, for example) and the interaction among the different flows at the network queues. The recovery time is, however, only affected by the transport protocol.

These results indicate the need for studying mechanisms that can better realize the target rates. We study a number of mechanisms in the following sections. We will look at three different levels of bandwidth allocation: (a) 60% of the available bandwidth is allocated, (b) 90% of the available bandwidth is allocated, and (c) Oversubscribed where the allocated bandwidth is 120% of the available bandwidth. With careful provisioning of the resources, the network should never operate in mode (c). However, without any mechanisms like RSVP [10] or path pinning, the flows with reservations could all go through the same link to result in oversubscription. We study this mode (c) as well to understand how well the different schemes work in such a situation.

Recently, simulation work on marking and dropping strategies has been done [2, 3, 4, 5, 6, 7]. These results indicated the need for better mechanisms for marking and dropping. Some of the conclusions of the earlier work include: (a) an application may not realize its reserved rate even when bandwidth is not oversubscribed [2, 4, 5], (b) flows with smaller RTTs may achieve higher bandwidth than flows with larger RTTs [2], (c) that realized rate is not proportional to the reserved rate when resources are plentiful [5]. These results provided strong motivation for our work reported here. Some of the earlier work [6, 7] has focused on similar issues in networks where marking strategies are different than the ones studies here. Our work considerably extends this earlier work and proposes new approaches to improving the realization of bandwidth guarantees.

We will consider two quantitative measures in evaluating various schemes. First, do the flows receive bandwidth corresponding to the reserved rates? Second, is the excess bandwidth fairly shared among all the flows? For an individual flow, without the knowledge of other flows, achieving the reserved rate is important. However, if the service provider cannot provide mechanisms to share the excess bandwidth fairly, the users may not perceive service differentiation with higher reserved rates and may not be willing to pay for higher-cost services.

The rest of the paper is organized as follows. Section 3 proposes a number of schemes to better achieve target rates in a differentiated services network. Section 4 presents a comparative evaluation of the proposed schemes. Section 5 presents an evaluation of the proposed schemes when flows may be aggregated. Section 6 presents conclusions and directions for further research.

3 Policies to achieve target rates

3.1 Limiting OUT packets

We first focus on the flows with the higher target rates. We noticed that in the earlier simulation, there were no IN packets dropped. A packet drop and a resulting contraction in sending rate by TCP resulted in realizing rates below the target rates. If a flow sent out few OUT packets, it is likely that this flow will not experience as many packet drops and hence may be able to realize the target rate. This policy aims to impact the number of packets dropped for each flow to better realize the target rates.

We modified the marker to send back information to the sender whenever one of its packets is marked OUT. The sender reduces the window by a packet as a result of this indication to avoid sending out any more OUT packets into the network. The result of this modification is shown in Figure 4. It is observed that the flows with higher target rates realize better throughput compared to the original RIO scheme. All the flows nearly achieved their reserved rates. However, the flows with higher targets do not come close to achieving their target rates at 60% level. At this subscription level, there is considerable excess bandwidth and the no-OUT mechanism doesn't try to capture any of the excess bandwidth since it prevents a flow from sending above its reserved rate. Thus, the excess bandwidth is fully captured by non-reserved best effort flows. This is clearly unacceptable since non-reserved flows get more bandwidth than other reserved flows. This scheme doesn't result in fair sharing of the excess bandwidth, and hence cannot provide proper service differentiation. At the 120% level, there is no excess bandwidth, and hence this scheme achieves rates very close to the targets. Limiting the OUT packets works well when the subscription level is high and doesn't work well at lower subscription levels.

3.2 Inverse-rate drop policy

This mechanism requires that every packet that is injected into the network is stamped with the service level besides the IN/OUT marking. The service level could be the reserved rate. The dropping policy is modified to take the service level into account. The higher the service level, the lower the probability for dropping a packet of that flow. Since each packet is marked with the service level, there is no need to maintain a state for each flow at the router. The rationale for this inverse drop policy (with respect to the service level) is that the flows at higher service level should get less packets dropped to counter the longer recovery times of the flows with higher target rates. This mechanism requires modifications to the marking and dropping policies.

The realized throughput of a TCP flow [11, 18, 19] is given by

$$t \leq 1.2 * B/(RTT * \sqrt{p}), \qquad (3)$$

where B is the packet size of the flow, RTT is the round-trip-time and p is the drop probability. Consider two flows with different target rates t_1 and t_2 with the same packet size and RTT. Then, based on the above equation

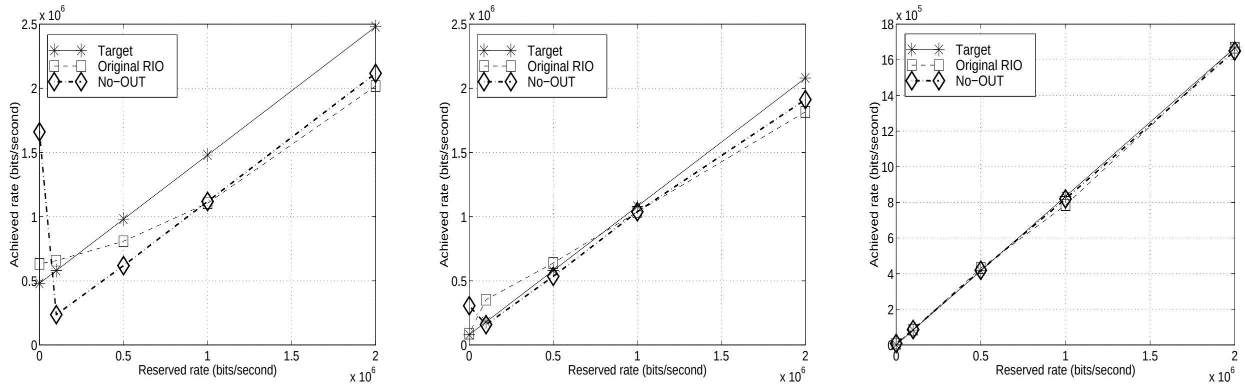
$$t_1/t_2 \;=\; \sqrt{(p_2/p_1)}. \qquad (4)$$

When we compare the number of packets dropped in a unit of time

$$d_1/d_2 \;=\; (t_1 * p_1)/(t_2 * p_2) \;=\; \sqrt{(p_1/p_2)} \;=\; t_2/t_1. \qquad (5)$$

This indicates that the number of packets dropped in a unit time should be inversely proportional to the target rate of that flow.

In a differentiated services network, a flow consists of IN packets and OUT packets. The rate of IN packets corresponds to the reserved rates and the rate of OUT packets correspond to the share of excess bandwidth. Ideally, the rate of OUT packets are the same for any two flows when excess bandwidth is equally shared. Also, if the RIO parameters are chosen care-



(a) 60% (b) 90% (c) 120%

Figure 4: No-OUT scheme at different subscription levels.

fully and if the network is not oversubscribed, the packet drops will all be OUT packets. So, the number of packets dropped in a unit time based on only the OUT packets of each flow is given by

$$d_1/d_2 \;=\; (OUT_1 * p_1)/(OUT_2 * p_2) \;=\; p_1/p_2, \quad (6)$$

where OUT_1 and OUT_2 correspond to the rate of OUT packets for flows 1 and 2 respectively and $OUT_1 \;=\; OUT_2$ in ideal situations. From the above two equations, it is clear that the drop probabilities should be inversely proportional to the target rates in a differentiated services network where most of the drops are OUT packets, i.e.,

$$p_1/p_2 \;=\; t_2/t_1. \quad (7)$$

The marker is unaware of the excess bandwidth and hence cannot easily determine the target rates. Hence, we use the reserved rates for marking the packets instead of the target rates. This analysis gives an idea of how dropping policy can be modified to counter the TCP congestion avoidance to better realize the target rates. The advantage of this scheme is that it doesn't require modifications to the TCP layers and hence can work with the existing TCP software.

Consider a non-responsive UDP flow sending data at rate s_i, with a reserved rate of r_i and a target rate of t_i. Then, the excess packets above $t_i = s_i \;-\; t_i$ should ideally be dropped from this flow [20]. The drop probability for this flow then should be

$$p_i \;=\; (s_i - t_i)/s_i \;=\; (1 - t_i/s_i) \quad (8)$$

This also shows that the drop probability should be inveresely related to the target rate of the flow (for a given sending rate), i.e., the higher the target rate, the lower the drop probability should be.

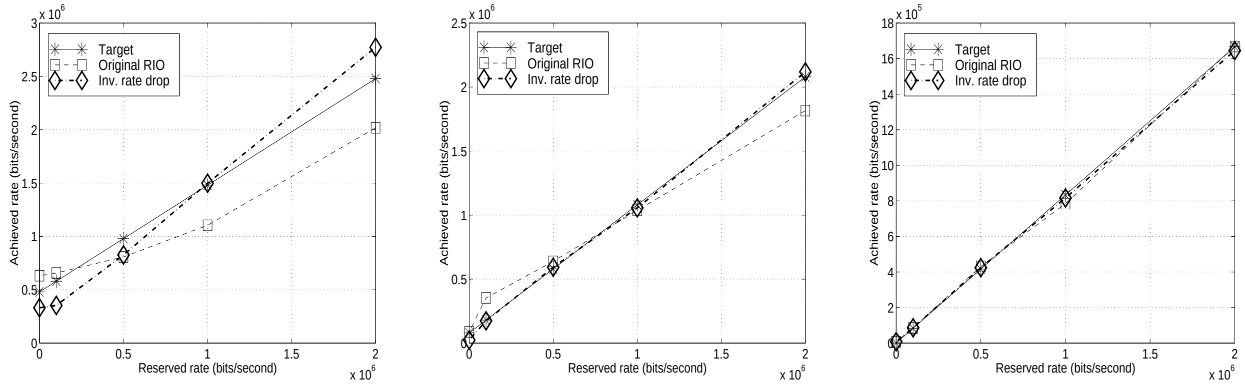
We used the following equation to calculate the drop probability of a flow with a reserved rate of r_i

$$p_i \;=\; k/(mk + r_i/r_{min}), \quad (9)$$

where k, m and r_{min} are suitably chosen constants. We chose r_{min} to be 0.1 Mbs, the smallest reservable rate in our simulations. When $r_i = 0$, $p_i = 1/m$. Hence, m can be chosen based on the target drop probability for a flow with zero reservation i.e., best-effort flow. Similarly, the parameter k can be chosen based on the target drop probability required for the flow with r_{min} reservation. We chose $k = 4$ and $m = 2$.

Figure 5 shows the results of this mechanism at different subscription levels. Most of the flows achieved their reserved rates better than in the original RIO scheme. The results also indicate that this inverse-drop policy achieved rates fairly close to the target rates. However, this scheme has a bias towards the higher rates. Similar trends in performance are observed at different subscription levels. Target rates are nearly achieved with a bias toward the higher rate flows.

To verify the sensitivity of this approach to the parameters chosen, we ran the simulations with different sets of target rates with the same parameters of k, m and r_{min}. The results had similar trends. The higher rate flows achieved targets better than unmod-



(a) 60%

(b) 90%

(c) 120%

Figure 5: Inverse-rate drop scheme at different subscription levels.

ified RIO scheme. Due to space limitations, those results are not being presented here.

3.3 Three Drop precedences

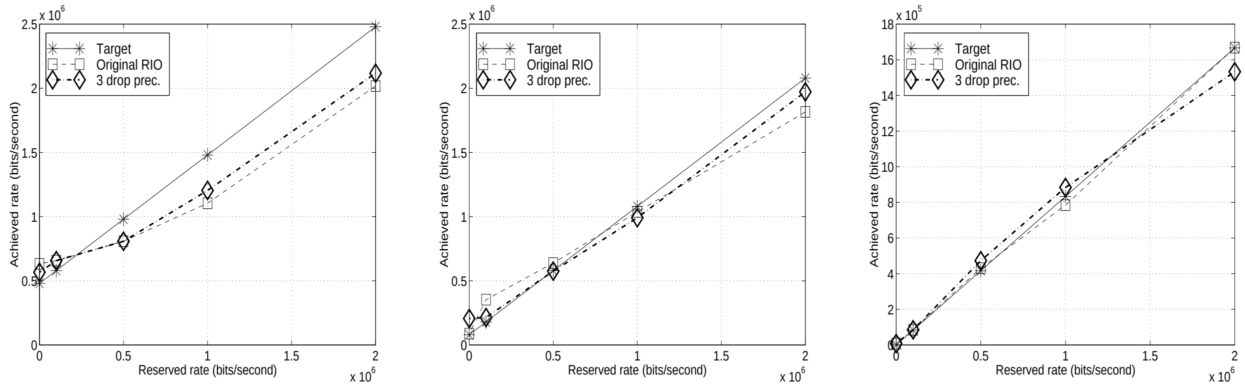
This approach uses three drop precedences, IN, OUT-IN and OUT-OUT. The marker continues marking packets IN when they conform to the leaky bucket profile. Instead of marking the remaining packets simply as OUT, it separates them into two categories. The marker keeps track of the long-term sending rate of the sender. If the long-term sending rate is higher than the reserved rate, more number of packets are marked OUT-OUT and if the long-term sending rate is lower than the reserved rate, more number of packets are marked OUT-IN. The router drops OUT-OUT packets earlier than the OUT-IN packets to give preference to responsive flows. This enables the marker to give a better treatment to flows that are falling behind in realizing their reserved rates.

Ideally, realized throughput of the sender should be used as a long-term measure. The acknowledgments may reach the sender through a different path without going through the marker. Also, if the sender is sending different sized packets, the marker would have to be aware of the size to estimate the goodput correctly even if the acknowledgments are going through the marker. To minimize these problems, we simply use the sending rate of the sender. As a result, for flows that don't respond to congestion, most of the packets above the reserved rate will be marked OUT-OUT. Flows that respond to congestion may have sending rates below the reserved rates and hence receive better treatment at the bottleneck link since more of their packets (above the reserved level) are marked OUT-IN.

Long-term sending rates are calculated by the same scheme as the scheme used to mark packets IN or OUT, but with a longer time window of 10 seconds. For the simulations, we use RED parameters 20/40/0.5 for OUT-OUT, 30/50/0.5 for OUT-IN and 50/100/0.02 for IN packets.

This approach requires modification to the marker and the router. The router deals with three drop precedences rather than two precedences in the other schemes. The marker also has to keep state information for each flow. However, the state information need only be kept at the ingress router for each flow and not at every router the flow passes through in the network. Earlier schemes we discussed did not have a need to maintain any state information for the flows. This scheme requires no changes to the sender.

The results of the simulations are shown in Figure 6. Most of the flows achieved their reserved rates better than in the original RIO scheme. It is also observed that this scheme realizes rates fairly close to target rates. However, we note that the flows with smaller targets exceed their targets and the flows with higher targets don't reach their targets. Similar trends are observed at different subscription levels. The target rates are nearly achieved with a bias toward the smaller flows. At lower subscription levels, all the flows send data above the reserved rates and hence the OUT packets are marked mostly OUT-OUT reducing the scheme to a two-drop precedence scheme.



(a) 60%

(b) 90%

(c) 120%

Figure 6: Three drop precedences at different subscription levels.

3.4 Two-windows TCP

Since the unmodified TCP protocol is unaware of reservation rates, congestion avoidance mechanisms could not protect the reserved rate of the flow. To avoid this problem, the congestion window of TCP is broken up into two pieces, the reserved window, $rwnd$, and the excess bandwidth window $ewnd$ such that $cwnd = rwnd + ewnd$. The reservation window $rwnd$ is obtained by multiplying the RTT of the flow with its reserved rate r_i. TCP is then modified to only reduce the $ewnd$ by half when an OUT packet is dropped and $cwnd$ as a whole is reduced only as a response to an IN packet loss. This requires that the sender keep track of how his packets are marked IN/OUT and then determine the congestion avoidance mechanism based on the dropped packet's marking. An OUT packet drop is considered as an indication of oversubscription of the excess bandwidth and an IN packet drop is considered an indication of oversubscription of link bandwidth. The modified congestion avoidance algorithm is shown in Figure 7. This scheme requires modifications to the transport protocol. It also requires that the sender be informed of IN/OUT markings of packets. This can be achieved by integrating the marker with the sender or by the marker informing the sender of the markings. A similar scheme has been recently studied in [6] with a slightly different marking strategy. We include this scheme here for completeness and to compare the other schemes with this approach.

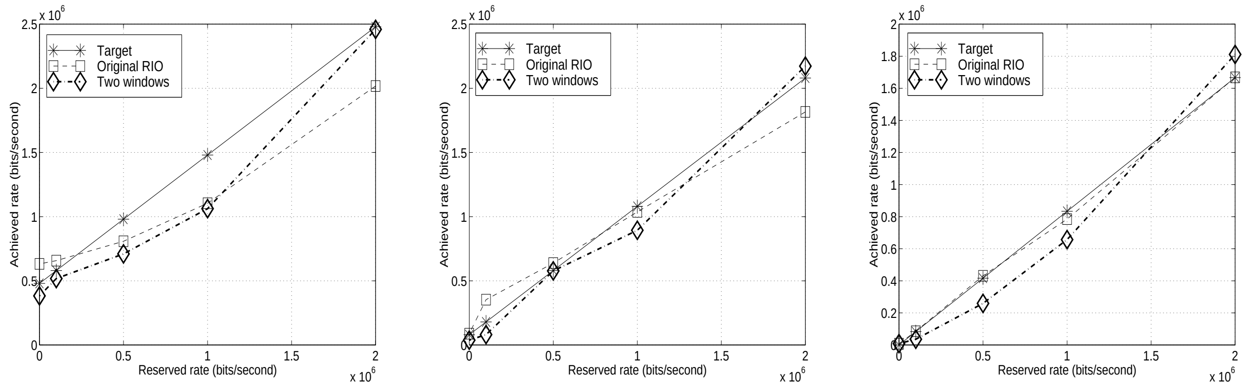
The results of the simulations are shown in Figure 8. It is observed that this scheme also achieves rates better than the original RIO algorithm. At 60% and 90% subscription levels, the flows with smaller target rates exceed their targets and the flows with the higher targets don't reach their target rates. At 120% subscription, the flows with higher target rates are favored.

```
After every packet loss detected

   if(OUT packet loss){
      rwnd = rtt * reserved_rate;
      if(rwnd < cwnd){
         ewnd = cwnd - rwnd;
         cwnd = rwnd + ewnd/2;
      }
   }
   else{               // IN packet loss
      cwnd = cwnd/2;
   }
```

Figure 7: Modified congestion avoidance algorithm

The throughput of each flow can be divided into IN packets corresponding to the reserved rate and OUT packets corresponding to the shared excess bandwidth. After an OUT packet drop, the time to recover to the original congestion window is inveresely proportional to the window size because of TCP's fast recovery mechanism (where congestion window is increased by one only after all the packets in the window are acknowledged). Hence, the flows with smaller target rates send more OUT packets than the flows with higher target rates and get more excess bandwidth. However, the IN packet throughputs are hardly affected by the target rates because the congestion window size is maintained to be larger than the reserved window. Thus, at lower subscription levels, even though the flows achieve their reserved rates, the flows with higher target rates cannot reach their target



(a) 60%

(b) 90%

(c) 120%

Figure 8: Two-windows TCP at different subscription levels.

rates. TCP's burstiness was observed to cause packets to be marked OUT even when there is sufficient bandwidth leading to discrepancies in the realized throughputs. Timer based transmission of packets could reduce the burstiness to increase the realization of target rates [6]. Suggested modifications (two-windows, use of timers) may make TCP more aggressive than current versions of TCP. Also, modifications to widely deployed software may be harder than modifications to the new network hardware to be deployed.

4 Comparison of all the schemes

Table 1 compares all the studied schemes against each other. For each scheme, we compute the realized utilization of the link. We also computed the mean-square-error (MSE) of the target rates and the realized rates. Since No-OUT scheme cannot let the flows with reservations share the excess bandwidth, we will not consider this scheme further, even though it shows good results at higher subscription level.

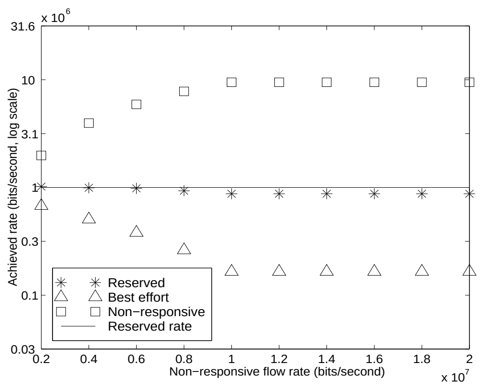
The results show that all the considered schemes do better at realizing reserved rates than the original RIO scheme. The new schemes (except for No-OUT) also achieve better realization of target rates (observed by MSE). The mean-square error of each scheme increases as the subscription level is decreased. At 60% subscription level, there remains 40% excess bandwidth. The difficulty in sharing this excess bandwidth among all the sources causes the achieved rates to diverge from the target rates. The results of unmodified RIO and three-drop schemes have the same trend that the flows with lower reserved rates get more than their target rats, and the flows with higher reserved rates cannot reach their target rate. It is because of two fundermental reasons. First is the TCP congestion avoidance mechanism. TCP was designed to share bandwidth equally (the additive increase results in fairer share [8, 17]), and thus the flows having more bandwidth lose more when congestion occurs. The second reason is that the profile marker does not know the target rates and uses reserved rates. In case of higher subscription level, the target rates and the reserved rates are not much different, and each source gets close to the target rates. The results of inverse-rate drop and two windows schemes have opposite trend to the results of unmodified RIO and three-drop schemes. The divergence of the results of inverse-rate drop scheme from the target rates is due to dropping packets based on reserved rates instead of target rates. At lower subscriptions, this results in larger errors.

4.1 Impact of a non-responsive source

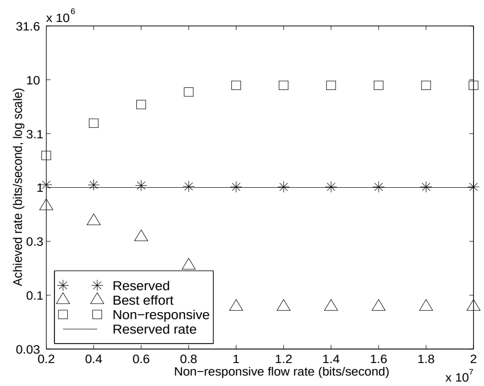
A source not responding to congestion, ideally should not take bandwidth away from sources that respond to congestion. Otherwise, a non-responsive flow can disturb the throughput guarantees of responsive multimedia flows. Figure 9 shows the impact of a non-responsive source with no reservation on the remaining sources. The simulation consisted of 10 TCP sources at 1 Mbps subscription and 10 best-effort sources along with a non-responsive UDP source at varying rates on a 20Mbps link. As the non-responsive source rate is increased, the reserved TCP flows start losing their reserved bandwidth in the unmodified RIO. How-

Table 1: Comparison of all the schemes

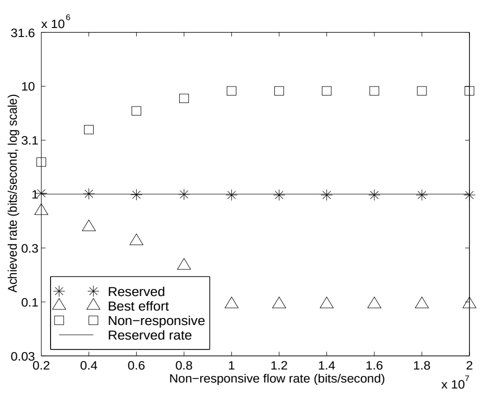
Subscript. Lvl.	0	0.1	0.5	1.0	2.0	Util.	MSE
			60%				
Target Rate	0.48	0.58	0.98	1.48	2.48	100%	0
Original RIO	0.63	0.66	0.80	1.10	2.01	87%	0.29
Inv. Rate Drop	0.33	0.35	0.83	1.49	2.77	96%	0.19
No-OUT	1.66	0.24	0.62	1.12	2.12	96%	0.61
Two Windows	0.38	0.52	0.71	1.06	2.46	86%	0.23
Three drop prec.	0.56	0.65	0.81	1.20	2.12	90%	0.22
			90%				
Target Rate	0.08	0.18	0.58	1.08	2.08	100%	0
Original RIO	0.09	0.35	0.64	1.04	1.81	98%	0.15
Inv. Rate Drop	0.02	0.17	0.59	1.05	2.11	99%	0.03
No-OUT	0.30	0.16	0.53	1.04	1.91	99%	0.13
Two Windows	0.04	0.07	0.58	0.89	2.17	94%	0.10
Three-drop prec.	0.20	0.21	0.57	0.99	1.97	99%	0.09
			120%				
Target Rate	0.00	0.08	0.42	0.83	1.67	100%	0
Original RIO	0.01	0.09	0.43	0.78	1.67	99%	0.03
Inv. Rate Drop	0.01	0.08	0.42	0.82	1.65	99%	0.01
No-OUT	0.01	0.09	0.42	0.82	1.65	99%	0.01
Two Windows	0.01	0.04	0.26	0.66	1.81	93%	0.12
Three-drop prec.	0.01	0.09	0.47	0.88	1.54	99%	0.06



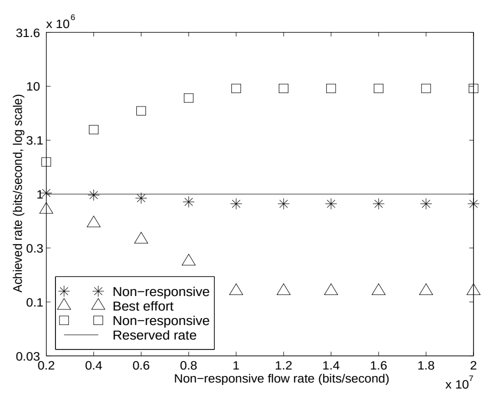
(a) Unmodified RIO



(b) Inverse-rate drop



(c) Three-drop precedences



(d) Two windows TCP

Figure 9: Impact of a non-responsive UDP flow

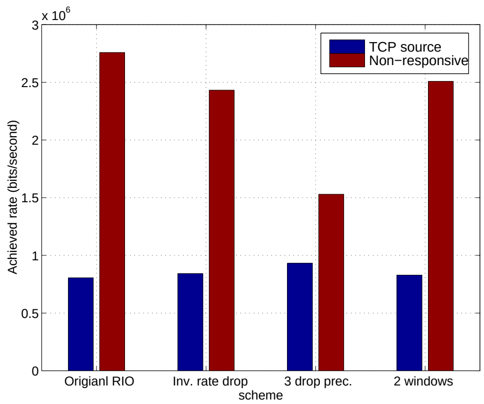


Figure 10: Impact of a non-responsive flow with reservation.

ever, the other schemes better protect the TCP flows with reservations. As the non-responsive source rate is increased, it realizes more and more bandwidth until it reaches a bandwidth of 10 Mbps. In inverse-drop and three-drop precedences schemes, the TCP reservations are protected, while the best-effort connections realize less and less bandwidth with increasing non-responsive source rates. In the unmodified RIO and the two-windows schemes, the reserved flows also get affected. The inverse-rate scheme and the three-drop precedence schemes provide better service differentiation between reserved flows and best-effort flows. In all the schemes, the non-responsive source gets contained around 10Mbps.

Figure 10 shows the impact of a non-responsive source with a reservation of 1 Mbps on 10 other TCP sources with reservations of 1Mbps. The link bandwidth is kept at 12 Mbps. The non-responsive source sending rate is kept at 6 Mbps. The non-responsive flow achieves better bandwidth in all the cases. The non-responsive flow achieved a rate of 2.76 Mbps in the unmodified RIO case, 2.4 Mbps in the inverse-rate drop scheme, 1.5 Mbps with three drop precedences and 2.5 Mbps in the two-windows scheme with a corresponding loss in the realized bandwidth of the TCP sources. All three proposed schemes contained the non-responsive flow better than the unmodified RIO scheme. The performance with three drop precedences is much better than the other schemes. It is also observed that all the schemes performed worse at reducing the impact of a non-responsive flow when the non-responsive flow has a reservation. This indicates that other schemes to identify (and possibly punish) non-responsive flows are required in addition to the marking and dropping policies.

4.2 Impact of different RTTs

The bandwidth realized by TCP flows is sensitive to RTTs between the senders and the receivers. So far, our simulation experiments conidered flows with the same RTTs. How do the results get affected when different RTTs are considered. To understand this, we considered flows at RTTs of 20ms, 40ms, 60ms, 80ms, and 100ms. At each RTT, we considered three flows, one with a high reservation, the second flow with a lower reservation and a third flow that is best-effort for a total of 15 flows. In the first experiment, we considered reservation levels of 1Mbps and 0.5Mbps and a link bandwidth of 10 Mbps (with 7.5 Mbps allocated out of 10Mbps). In a second experiment, we considered the same flows on a link bandwidth of 30Mbps such that only 25% of the bandwidth is allocated.

The simulation results are shown in Table 2. Clearly, in all the schemes, the flows with smaller RTTs experienced better service than flows with longer RTTs. However, no scheme seems to have a clear advantage over the others. To quantify the differences, we used the following measures: (a) Mean Square Error (MSE) used earlier based on the achieved and target rate differences, (b) Utilization, (c) Fairness within a service class (best-effort, 0.5M reservation, 1M reservation) measured by the maximum rate divided by the minimum rate within the same reservation level, (d) Service differentiation measured by the minimum rate achieved at a particular reservation level divided by the maximum rate achieved at the next lower reservation level, and (e) Reservation success measured by counting the number of flows reaching the reserved rate. Since we considered three different service levels, we get three fairness values (one for each class) and two service differentiation values (min of 0.5M/max of BE, min of 1M/max. of 0.5M). Ideally, fairness values should equal 1 and the service differentiation values should be greater than 1. These quantitative measures are shown in Table 3 for both the experiments. For example, two-windows scheme achieve a fairness measure of 3.18 for 1 Mbps flows at 25% subscription i.e., a flow (with RTT of 20ms) achieved 3.18 times the bandwidth achieved by another flow with the same reserved rate of 1Mbps. Similarly, a service differentiation of 0.4 between 0.5 Mbps flows and 1 Mbps flows means that a flow with a reservation of 1 Mbps achieved only 40% of the bandwidth achieved by a flow with a reservation of only 0.5 Mbps due to differences in RTTs.

From the table, it is observed that proposed schemes do better at realizing reserved rates and provide bet-

Table 2: Throughputs of different schemes at different RTTs.

Scheme	20ms BE/0.5M/1M	40ms BE/0.5M/1M	60ms BE/0.5M/1M	80ms BE/0.5M/1M	100ms BE/0.5M/1M
75%					
Target rates	0.17/0.67/1.17	0.17/0.67/1.17	0.17/0.67/1.17	0.17/0.67/1.17	0.17/0.67/1.17
Original RIO	0.46/0.71/1.23	0.29/0.68/1.09	0.23/0.61/1.09	0.27/0.61/0.90	0.18/0.50/0.92
Inv. rate drop	0.06/0.65/1.65	0.08/0.64/1.41	0.06/0.58/1.17	0.02/0.55/1.19	0.01/0.55/1.20
Three drop prec.	0.43/0.69/1.19	0.28/0.70/1.12	0.25/0.62/1.02	0.19/0.52/1.01	0.20/0.53/1.02
Two windows	0.15/1.04/1.27	0.01/0.70/1.03	0.02/0.53/0.99	0.03/0.51/0.93	0.01/0.46/0.91
25%					
Target rates	1.50/2.00/2.50	1.50/2.00/2.50	1.50/2.00/2.50	1.50/2.00/2.50	1.50/2.00/2.50
Original RIO	3.09/3.17/3.37	1.66/1.93/2.20	1.18/1.50/1.71	1.02/1.03/1.40	0.96/1.14/1.40
Inv. rate drop	1.03/2.87/6.84	0.46/1.81/4.58	0.29/1.08/3.28	0.23/0.94/2.12	0.19/0.87/2.32
Three drop prec.	3.09/3.17/3.37	1.66/1.93/2.20	1.18/1.50/1.71	1.02/1.03/1.40	0.96/1.14/1.40
Two windows	1.98/3.22/4.01	1.42/2.02/2.66	0.93/1.63/1.82	0.73/1.12/1.56	0.63/1.10/1.26

Table 3: Performance summary of different schemes at different RTTs.

Scheme	MSE(Mbps)	Util.	Fairness (BE/0.5M/1M)	Serv. Differentiation (0.5M/BE)/(1M/0.5M)	Reservation success
75%					
Original RIO	0.14	98%	2.53/1.41/1.39	1.09/1.30	80%
Inv. rate drop	0.16	99%	8.43/1.17/1.37	8.85/1.84	100%
Three drop prec.	0.12	99%	2.11/1.31/1.11	1.21/1.54	100%
Two windows	0.18	87%	15.8/2.25/1.38	3.17/0.87	60%
25%					
Original RIO	0.83	90%	3.18/2.77/2.40	0.37/0.44	100%
Inv. rate drop	1.50	96%	5.37/3.28/2.95	0.84/0.80	100%
Three drop prec.	0.83	90%	3.18/2.77/2.40	0.37/0.44	100%
Two windows	0.82	87%	3.13/2.92/3.18	0.56/0.40	100%

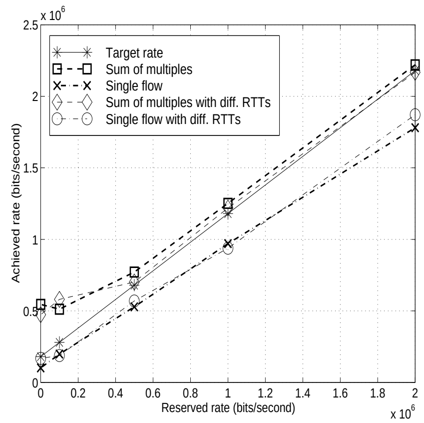
ter service differentiations. It is also observed that the MSE is higher at lower subscriptions for all the schemes, pointing again to the difficulty in assigning the excess bandwidth fairly. The three-drops scheme and the two-windows scheme have better mean square error from the target rates at 75% and 25% subscription level, respectively. They also have better fairness measures compared to the other two schemes. In three-drops and inverse-rate drop schemes, all flows reach their reserved rates. Inverse-rate drop scheme achieves the best service differentiation among the schemes considered.

5 Effect of aggregated sources

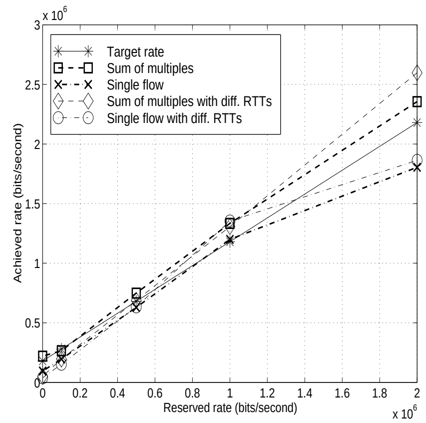
In this section, we present simulation results with aggregated sources. The motivation of these simulations is the fact that it may not be practical to reserve bandwidth and put a profile marker at each host in the real Internet. A profile marker may be assigned to a group of hosts (a university or a company, for example). In this situation, we have two issues. First issue is how does aggregation impact the guarantees of all the flows, aggregated or otherwise. The second issue is how to assign the reserved group bandwidth to each host in the group so that an individual application (such as IP telephony) can realize the necessary guarantees. In this paper, we focus on the first issue.

The simulation topology is the same as the topology used in the previous simulations shown in Figure 2 except that the odd-numbered sources (1, 3, 5, 7, and 9) now consist of an aggregation of three separate flows. Hence, at each reservation level, we have an individual flow and an aggregated flow. We used the same RED parameters and reservation rates, and the bottleneck bandwidth is set to 9 Mbps. Since a total of 7.2 Mbps is allocated, the subscription level is 80%. We ran two simulation experiments using this configuration. In the first experiment, we used the same RTT, 60 ms for all the 20 flows. In the second experiment, we used RTTs of 40 ms, 60 ms, and 80 ms for the three sources of an aggregated pool, and an RTT of 60 ms for single sources. Flows within an aggregated source may have different RTTs as they may be talking to different hosts after passing through the same bottleneck link.

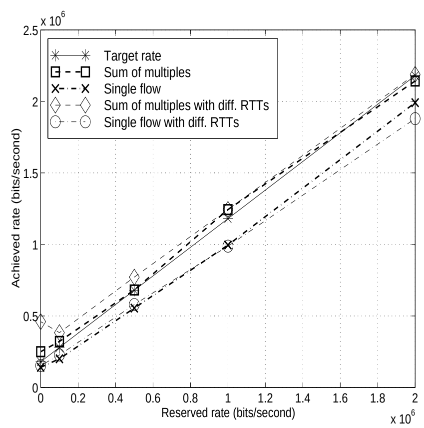
Figure 11 shows the results of the simulations. In all



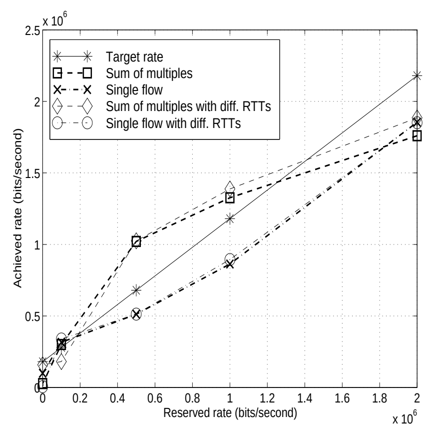
(a) Unmodified RIO



(b) Inverse-rate drop



(c) Three drop precedences



(d) Two windows TCP

Figure 11: Comparisons between aggregated sources and single source

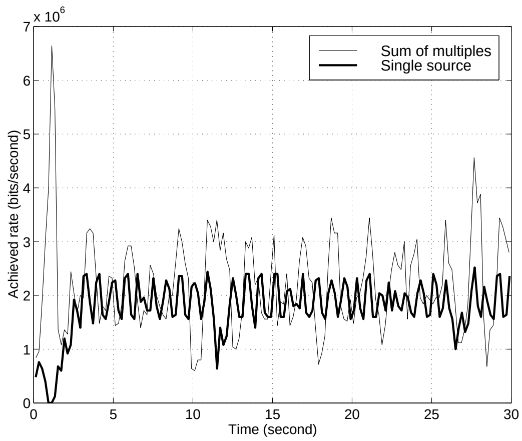


Figure 12: Measured rates vs. time of single and aggregated sources with 2Mbps subscription

the schemes, the results are not much different across the two experiments. This may indicate that aggregation may blunt the effect of RTT differences. In all the schemes, the aggregated sources realize higher throughputs than a single source. The three flows within an aggregated source claim three times as much of the shared excess bandwidth than a single source and hence the difference.

Figure 12 shows the bandwidth realized by two sources, one aggregated and one single source, both at 2 Mbps reservation. In the figure, the peak rates of the aggregated sources are much higher than the peak rates of the single source. It means that the aggregated sources get more excess bandwidth than the single source. The figure also shows that the measured rate of the single source fluctuates more frequently than the rate of aggregated sources. Since each of the aggregated sources reacts to congestion individually, the rate of the aggregated sources is smoother.

6 Conclusions

Multimedia applications require throughput guarantees for delivery over networks. In this paper, we have proposed and evaluated several schemes to improve the realization of throughput guarantees in the future internet that employs a differentiated services framework. The proposed schemes are shown to improve the realization of reserved rates and to provide better service differentiation than the original RIO scheme. The results also show that non-responsive sources can be controlled better by the proposed schemes. Among the proposed schemes, inverse-rate drop realized rates closer to target rates and provided better differentiation of services. Three drop precedences scheme has advantage in controlling a non-responsive sources and reducing the impact of differences in RTTs. Both these schemes do not require modifications to transport (TCP) layers. The two-windows scheme modifies TCP to provide similar benefits. All of the proposed schemes require enhancements to the basic RIO scheme, either at the marker, sender or in the network.

We also reported on extensive simulations on the sensitivity of the results to differences in RTTs of flows and aggregation of flows. Our results indicate that the provided guarantees are soft i.e., not realized in all the situations. It was observed that a non-responsive flow can disturb the throughputs of other flows. It was observed that differences in RTTs resulted in the realization of different rates even when flows had the same reservations. However, it was shown that the impact of differences in RTTs could be reduced by the aggregation of sources and fair sharing of bandwidth at the edge routers. Even though fair-sharing of excess bandwidth proved to be a challenge, the reservations could be mostly met.

Our future work will focus further on minimizing the throughput variations due to RTT differences, non-responsive flows and flow aggregation. We are also looking at the impact of network routing and routing changes on the realized throughputs. We are also studying the impact of different scheduling mechanisms at the edge routers on the realized throughputs.

References

[1] Hui Zhang and Domenico Ferrari, "Rate-Controlled Static-Priority Queueing," *Proceedings of IEEE INFOCOM'93*.

[2] J. Ibanez and K. Nichols, "Preliminary Simulation Evaluation of an Assured Service," draft-ibanez-diffserv-assured-eval-00.txt, INTERNET DRAFT, August, 1998.

[3] K. Nichols, V. Jacobson, and L. Zhang, "A Two-bit Differentiated Services Architecture for the Internet," draft-nichols-diff-svc-arch-00.txt, INTERNET DRAFT, December, 1997.

[4] David D. Clark and Wenjia Fang, "Explicit Allocation of Best-Effort Packet Delivery Service," November, 1997. http://diffserv.lcs.mit.edu/exp-alloc-ddc-wf.pdf

[5] Anindya Basu and Zheng Wang, "A Comparative Study of Schemes for Differentiated Services," *Bell labs Technical Report*, Aug. 1998.

[6] W. Feng, Dilip D. Kandlur, D. Saha, and Kang G. Shin, "Understanding TCP Dynamics in an Integrated Services Internet," *Proceedings of the International Workshop on NOSSDAV*, May, 1997.

[7] W. Feng, Dilip D. Kandlur, D. Saha, and Kang G. Shin, "Adaptive Packet Marking for Providing Differentiated Services in the Internet." *Proc. of Int. Conf. on Network Protocols*, Oct. 1998.

[8] Van Jacobson and Michael J. Karels, "Congestion Avoidance and Control," *Proceedings of SIGCOMM'88*, Stanford, CA, August, 1998.

[9] Network simulator (Ns), University of California at Berkeley, CA, 1997. Available via http://www-nrg.ee.lbl.gov/ns/.

[10] L. Zhang, S. Deering, D. Estrin, S. Shenker, and D. Zappala, "RSVP: A New Resource ReSerVation Protocol," *IEEE Transaction on Networking*, September, 1993.

[11] Sally Floyd and Kevin Fall, "Promoting the Use of End-to-End Congestion Control in the Internet," http://www-nrg.ee.lbl.gov/floyd/end2end-paper.html, Fubruary, 1998.

[12] A. Demers, S. Keshav and S. Shenker, "Analysis and simulations of a fair queuing algorithm," *Proc. of ACM SIGCOMM*, 1989.

[13] S. Golestani, "Duration limited statistical multiplexing of delay-sensitive traffic," *Proc. of INFOCOMM*, 1991.

[14] A. Parekh and R. Gallager, "A generalized processor sharing approach to flow control in integrated service networks: The multiple node case," *Proc. IEEE INFOCOMM'93*, Mar. 1993.

[15] B. Suter, T. V. Lakshman, D. Stiliadis and A. Choudhury, "Design considerations for supporting TCP with per-flow queuing," *Proc. of INFOCOMM '98*, Apr. 1998.

[16] Z. Wang and J. Crowcroft, "QOS routing for supporting resource reservation," *IEEE Journal on selected areas in communications*, 1996.

[17] D. M. Chiu and R. Jain, "Analysis of increase and decrease algorithms for congestion avoidance in computer networks," *Journal of Computer Networks and ISDN Systems*, 1989.

[18] J. Padhye, V. Firoiu, D. Towsley and J. Kurose, "Modeling TCP throughput: A simple model and its empirical validation," *Proc. of ACM SIGCOMM*, 1998.

[19] T. Ott, J. Kemperman and M. Mathis, "The stationary distribution of ideal TCP congestion avoidance," *ftp://ftp.bellcore.com/pub/tjp/TCPwindow.ps*, Aug. 1996.

[20] I. Stoica, S. Shenker and H. Zhang, "Core-Stateless Fair queueing: Achieving approximately fair bandwidth allocations in high speed networks," *Proc. of ACM SIGCOMM*, Oct. 1998.